

文章编号:1673-0062(2011)03-078-05

变化数据捕获研究及基于 SQL SERVER 的开发与应用

王丽君,李 萌,徐卓然,雷龙艳,周 倩,阳小华

(南华大学 电气工程学院,湖南 衡阳 421001)

摘 要:变化数据捕获是数据集成的核心内容,为高效地实现数据集成,探讨了变化数据捕获的常见机制,重点介绍了 SQL SERVER 的变化数据捕获方案——更改跟踪和变化数据捕获,并以实际开发项目为背景,介绍在偶尔连接场景下,如何运用变化数据捕获技术,合理设计数据集成方案,满足业务要求,实现异构数据源的有效更新.
关键词:变化数据捕获;更改跟踪;变更数据捕获
中图分类号:TP31 **文献标识码:**B

Research on the Change Data Capture and Its Application in SQL SERVER Database

WANG Li-jun, LI Meng, XU Zhuo-ran, LEI Long-yan, ZHOU Qian, YAN Xiao-hua
(School of Electric Engineering, University of South China, Hengyang, Hunan 421001, China)

Abstract: The change data capture is the core content of data integration. For implementing data integration efficiently, this paper introduced the usual methods of change data capture, laid heavy stress on the change data capture solutions of SQL SERVER and demonstrated how to use it with an application example.
key words: change data capture; change tracking; changing data capture

随着信息化建设的不断发展,各类信息系统的数量快速膨胀,在提供丰富数据源的同时,也引发了信息孤岛问题,使得这些数据资源无法共享、相互利用,更无法满足从数据到信息再到知识的应用,企业应用集成 EAI 应运而生,它包括多个层面的集成,其中数据集成最为基础也最为普遍,数据集成通常采用 ETL 对各类异构数据源进行抽取、转换和清洗. 数据集成的核心内容就是变化数据捕获,由于信息系统大多属于在线联机事务处理 OLTP,因此其中的数据不断变化,又因为信息系统多采用数据库管理系统 DBMS 管理数据,如常见的 ORACLE、SQL SERVER、IBM DB2、MY SQL 等,所以需要充分利用 DBMS 特性,及时捕获变化的数据,实现数据更新与同步,为数据集成奠

收稿日期:2011-07-12
基金项目:湖南省教育科学“十一五”规划基金资助项目(XJK08BGD017);南华大学 2010 高等教育研究与改革基金资助项目(2010ZC011)
作者简介:王丽君(1979-),女,湖南邵阳人,南华大学电气工程学院实验师,硕士. 主要研究方向:ETL、EDA、DSP 研究.

定坚实基础.

论文首先探讨了常见的变化数据捕获机制,然后介绍 SQL SERVER 2008 提供的变化数据捕获方案,即更改跟踪(Change Tracking)与变更数据捕获(Changing Data Capture).最后以实际项目中的数据更新模块为例,说明如何利用 SQL SERVER 方案进行开发与实现.

1 变化数据捕获机制

研究者们对变化数据捕获机制进行了深入研究,目前的主题可分为两类:变化数据捕获机制和特定环境下的应用.2002 年,者敬在其博士论文中将变化捕获方法归纳为六种:快照法、触发器法、日志法、API 法、影子表法和控制表变化法,并从捕获方法、更新方式、适用范围等九个方面对它们进行综合比较^[1];2004 年,陆剑峰等结合开发示例,介绍利用 ORACLE 9i 的 CDC(Changing Data Capture) 机制解决数据增量更新的方法^[2];2008 年,杨鹏等在分析现有实现方案不足的基础上,提出一种基于触发器的高效通用变化数据捕捉方法,设计实现了跨 DBMS(Oracle, Sybase 和 MS SQL Server)的相应同步策略^[3];2009 年,戴浩等将 ETL 中数据增量抽取的常见机制归纳为 7 种:触发器、时间戳、全表删除插入、全表比对、日志表、系统日志分析和特定数据库,对这些机制的原理、条件、方法以及运行效率等方面进行了详细的阐述,并从兼容性、完备性、性能和侵入性四个方面分析和比较了各种数据增量抽取机制的优劣性^[4];2010 年,陈东亮等提出了一种基于任务的控制表法,该方法在控制表法的基础上增加了任务和类型等相关信息,同时以净变化量提取组件作为辅助,能够更有效地适用于多节点间异构数据库的数据交换,并给出了在变化捕获方法中因使用触发器而带来的数据震荡问题的解决方法^[5].

2 SQL SERVER 的变化数据捕获机制

变化数据捕获的两个的典型应用场景是:支持数据仓库的更新,以及支持异构、偶尔连接的系统进行同步处理^[6]. SQL SERVER 2008 提供了两种变化数据捕获机制^[7],即更改跟踪(Change Tracking)与变更数据捕获(Changing Data Capture),Paul 对两者进行深入比较^[6].

2.1 更改跟踪

更改跟踪采用同步机制,只能跟踪表中已更

改的特定数据行(或者经过更改的数据列).这是为了解决偶尔连接的系统方案而设计的,图 1 说明了使用更改跟踪的数据同步处理.

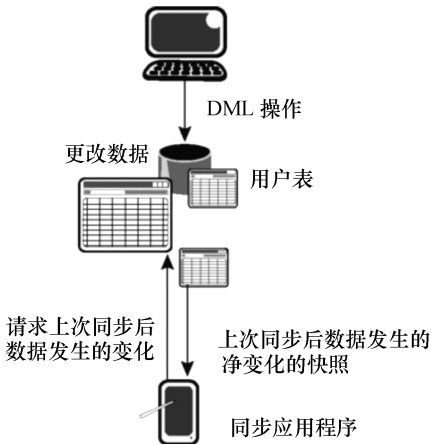


图 1 使用更改跟踪的偶尔连接的系统
Fig. 1 An occasionally connected system using change tracking data

2.2 变更数据捕获

变更数据捕获使用的是异步机制,可以跟踪表(或是表中一组定义的数据列)发生的所有更改,它专为数据仓库 ETL 过程等情形设计的.图 2 说明了不同时间段获取的更改数据.变更数据捕获机制会将更改的数据提取到一组表,最新的更改在表的最上方.然后,ETL 过程对存储变更数据的表查询在固定时段内发生的所有更改,允许 ETL 过程限制每批获取的数据量.

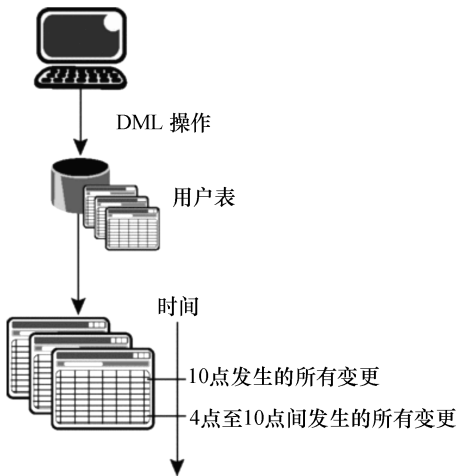


图 2 不同时间段获取的历史更改数据
Fig. 2 Historical change data being consumed in time slices

这两种机制与自定义解决方案的相同之处,由于必须在某个位置存储更改数据,因此都会增加 I/O 和记录;两者区别在于,用于存储更改数据的表必须与要跟踪的表位于相同的数据库中. 这说明所有更改数据都将包含在备份中,从而可通过日志传送或数据库镜像在网络上传输.

就开发而言,这两种机制可以明显降低变化数据捕获的复杂性. 1) 因为无论是哪一种技术,都不需要更改表架构或触发器;2) 两者都具有可配置的自动清除过程,可依据事务提交时间对更改排序,并且提供内置函数来检索更改信息.

3 应用实例

本应用实例是某市公安局社区警务系统中数据更新模块. 数据来自省厅的人口数据仓库,省厅 DBMS 采用 ORACLE,市局使用 SQL SERVER,两者为异构数据源,采用单向增量抽取,即从省厅更新到市局,具体要求:1) 添加市局没有的记录,2) 对双方均存在的记录,只更新部分字段的内容,3) 被市局修改过的字段,其内容不允许更新. 该模块的两个核心内容为定位实体和捕获变化数据,捕获粒度精确到列,具体业务逻辑如图 3 所示.

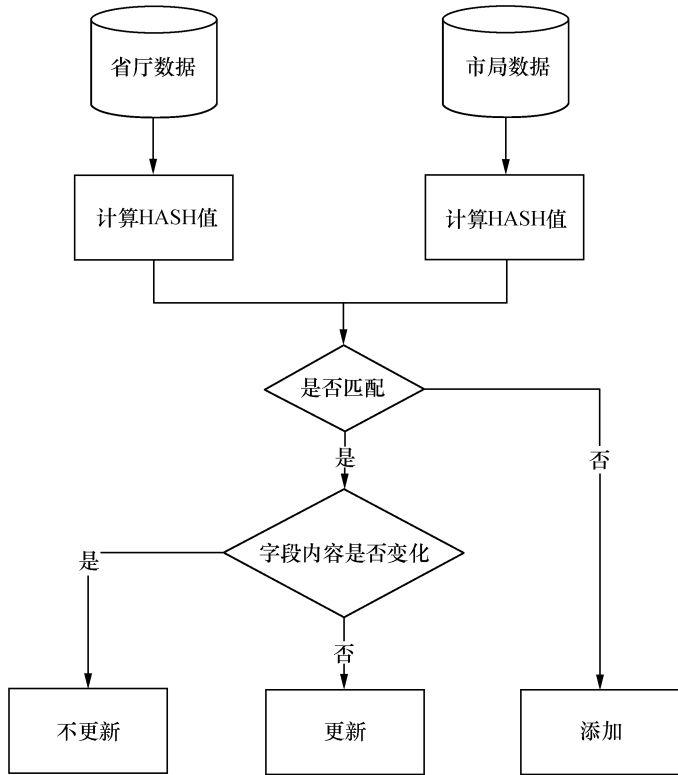


图 3 项目业务逻辑
Fig.3 Project business logical

3.1 定位实体

由于数据源和目标异构,两者没有统一的实体标识,为准确地更新数据,必须为每条记录建立唯一的主键,又因为人口数据量大,还必须减少主键值碰撞的几率,所以项目中采用 HASH 函数 SHA1 生成主键.

人口数据字段众多,如果将所有字段均作为 HASH 函数的输入,虽然保证了主键的惟一性,但是将大幅增加计算量,严重影响性能,因此,项目中采用候选键作为 SHA1 的输入,极大缩减了计

算代价,减轻对系统性能的影响.

3.2 捕获变化数据

因为数据抽取的时间间隔较长,通常为 1~4 周进行一次,属于典型的偶尔连接场景,所以项目中采用更改跟踪(Change Tracking)机制. 同时,业务只关心特定列是否存在变化数据,并不在意变化的具体内容,因此需要启用列跟踪支持. 利用更改跟踪来捕获变化数据可分为两步:首先启用并配置更改跟踪,然后使用更改跟踪函数获取更改^[7].

因保密需要,采用 AdventureWorks 作为示例数

数据库,库中建有一张表 Product,结构如表 1 所示.

表 1 Product 表结构

Table 1 Construction of table product

字段名称	是否为空	数据类型
Id	NOT NULL	INT
Name	NULL	NVARCHAR(100)
Price	NULL	NUMERIC(8,2)

1) 启用并配置更改跟踪

对数据库启用更改跟踪的执行脚本如下:

——对数据库启用更改跟踪

ALTER DATABASE AdventureWorks

SET CHANGE_TRACKING = ON

——开启自动清理,数据保存期为两天

(AUTO_CLEANUP = ON, CHANGE_RETENTION = 2 DAYS)

——对表启用更改跟踪

ALTER TABLE dbo. Product

ENABLE CHANGE_TRACKING

——跟踪列更新

WITH(TRACK_COLUMNS_UPDATED = ON)

为表配置了更改跟踪后,任何影响该表中的行的 DML 语句都将被记录下来,SQL SERVER 数据库引擎会将有关哪些列已更新的额外信息存储到内部更改跟踪表中. 使用更改跟踪可以获取与每个行所做更改相关的信息,如:导致更改(插入、更新和删除)的 DML 操作的类型或更新操作所更改的列.

2) 获取更改

SQL SERVER 提供了五个更改跟踪函数,见表 2,用于记录应用于被跟踪表的插入、更新和删除操作,同时采用易于使用的关系格式提供有关更改的详细信息.

表 2 更改跟踪函数

Table 2 Change tracking functions

函 数	说 明
CHANGETABLE(CHANGES)	返回自指定版本起对表所做的所有更改的跟踪信息.
CHANGETABLE(VERSION)	返回指定行的最新更改跟踪信息.
CHANGE_TRACKING_MIN_VALID_VERSION()	返回使用 CHANGETABLE 函数时从指定表获取更改跟踪信息所使用的最低有效版本.
CHANGE_TRACKING_CURRENT_VERSION	获取与上次提交事务关联的版本. 可以在下次使用 CHANGETABLE 枚举更改时使用此版本.
CHANGE_TRACKING_IS_COLUMN_IN_MASK	用于解 释 由 CHANGETABLE (CHANGES …) 函 数 返 回 的 SYS _ CHANGE _ COLUMNS 值.
WITH CHANGE_TRACKING_CONTEXT	用于在应用程序更改数据时指定更改上下文,例如发起方 ID.

查询数据更改的脚本如下:

```
SELECT CT. Id, P. Name, P. Price
,CHANGE_TRACKING_IS_COLUMN_IN_MASK
(
    COLUMNPROPERTY ( OBJECT_ID ( '
Product' ), 'Name', 'ColumnId' )
,CT. SYS_CHANGE_COLUMNS
) AS NameIsChanged
,CHANGE_TRACKING_IS_COLUMN_IN_MASK
(
    COLUMNPROPERTY ( OBJECT_ID ( '
Product' ), 'Price', 'ColumnId' )
,CT. SYS_CHANGE_COLUMNS
) AS PriceIsChanged
```

```
,CT. SYS_CHANGE_VERSION
, CT. SYS _ CHANGE _ OPERATION, CT. SYS _
CHANGE_COLUMNS
FROM Product AS P
RIGHT OUTER JOIN
CHANGETABLE ( CHANGES Product, @ last _syn-
chronization_version ) AS CT
ON P. Id = CT. ID
```

查询更改跟踪结果如图 4 所示,其中 SYS_CHANGE_OPERATION 有三种取值:I、U、D,对应的 DML 操作为 insert、update 和 delete, SYS_CHANGE_COLUMNS 为受 update 影响的字段编码,SYS_CHANGE_VERSION 表示 DML 的操作顺序,图 3 可解读为:先更新记录 Id = 1 的 Price 字

段,然后更新记录 Id = 5 的 Name 字段,接下来插入记录 Id = 7,最后删除 Id = 6 的记录.

Id	Name	Price	NamelsChanged	PricelsChanged	SYS_CHANGE_VERSION	SYS_CHANGE_OPERATION	SYS_CHANGE_COLUMNS
1	Apple	4.50	0	1	2	U	0x0000000003000000
5	Grapes	15.00	1	0	3	U	0x0000000002000000
6	NULL	NULL	1	1	5	D	NULL
7	StrawBerry	25.00	1	1	4	I	NULL

图 4 更改跟踪结果

Fig. 4 Result of change tracking

需要注意的是,启用了更改跟踪的数据库具有一个版本计数器,在对启用了更改跟踪的表进行更改时,该计数器会随之递增. 每个更改的行都有一个关联的版本号. 将请求发送到应用程序以查询更改时,将调用一个函数以提供版本号. 该函数返回在该版本之后所做的所有更改的相关信息. CHANGE_RETENTION 指定的时间决定了所有应用程序必须每隔多长时间从数据库中请求一次更改. 如果应用程序使用的版本值早于表的最低有效同步版本,将无法执行有效的更改枚举,必须重新初始化所有数据.

4 结 论

变化数据捕获是数据集成的核心内容. 由于数据源结构各异,业务要求千差万别,故实现方案的设计需要灵活应对. 论文探讨了变化数据捕获的常见机制,并结合实例说明了 SQL SERVER 的更改跟踪机制,它较好的解决了偶尔连接场景下的变化数据捕获问题.

参考文献:

[1] 者敬. 开放式异构数据库复制框架的研究与实现 [D]. 北京:中国科学院软件研究所,2002;19-23.

[2] 陆剑峰,张浩. 数据仓库数据更新的研究与基于 Oracle数据库的开发与应用[J]. 计算机工程与应用, 2004(26):168-170.

[3] 杨鹏,杨海涛,王正华. 异构数据库变化捕捉及同步策略[J]. 计算机工程,2008,34(16):53-59.

[4] 戴浩,杨波. ETL 中数据增量抽取机制研究[J]. 计算机工程与设计,2009,30(23):5552-5555.

[5] 陈东亮,孙静. 数据交换系统中变化捕获方法的研究与实现[J]. 计算机工程与设计,2010,31(1):94-97.

[6] Paul S Randal. Tracking Changes in Your Enterprise Database [J/OL]. Microsoft TechNet Magazine, 2008, [2008-11-28]. <http://technet.microsoft.com/en-us/magazine/2008.11.sql.aspx>.

[7] Tracking Data Changes[J/OL]. Microsoft MSDN library, 2011, [2011-02-11]. [ttp://msdn.microsoft.com/en-us/library/bb933994.aspx](http://msdn.microsoft.com/en-us/library/bb933994.aspx).